Single Event Multiple Transient Analysis of Digital Circuits using Satisfiability Modulo Theories

Prof. Otmane Ait Mohamed

ECE Department, Concordia University, Montreal, Canada

UNIVERSITÉ Concordia UNIVERSITY

Concordia University Hardware Verification Group

Seressa'2022, CERN, Geneva, December 5-9, 2022

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

- Ghaith Bany Hamad, Ph.D. Concordia 2017

  Senior Formal Verification Engineer, Nvidia, California, USA
- Ghaith Kazma, Masc. Concordia 2017

  Machine Learning Engineer Machine Learning Engineer, Google
- Sowmith Nethula, Masc. Concordia 2019

  Associate Digital Verification Engineer, Huawei, Ottawa, Canada
- Vivek Bansal, Masc. Concordia 2022

  Digital Verification Engineer, XtremeEDA (Accenture)

Outline

Motivation

Proposed Methodology

- Fault modeling in SMTs
- Fault propagation analysis using SMTs
- Layout based SEMT fault extraction

Results

Summary

Motivation

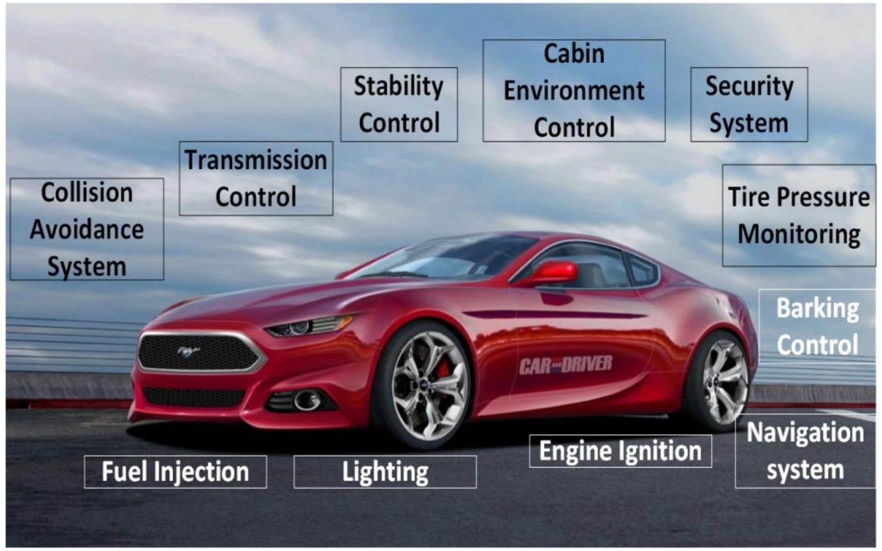
- Digital designs reliability is a major concern for chip designers
- Soft errors induced by radiations have become one of the most challenging issues that impact reliability
- Develop fault tolerant design methods
- ➔ Fast, accurate, and efficient analysis and estimation techniques of soft error in modern-DSM technologies

Motivation

Aerospace/Military

Medical Devices

Networking



Automotive

SEEs estimation

Three main ways to analyze SEEs:

1. Hardware testing (beam testing/laser testing)
   - (+) Most realistic and accurate
   - (-) Requires finished implementation
   - (-) May damage the device
   - (-) Costly

2. Fault injection (emulation/simulation)
   - (+) Less accurate than hardware testing but still very useful
   - (-) Test time grows with possible test cases
   - (-) Analysis of all masking effects (logical, electrical, and time window masking)
   - (-) The accuracy is related to the size of the simulated sample over the total vector space size
   - (-) Simulation cannot guarantee the design is completely “verified”

     ➔ Not acceptable in safety-critical applications

Formal verification

The application of rigorous, mathematics-based techniques to establish the correctness of computerised systems

Main techniques:

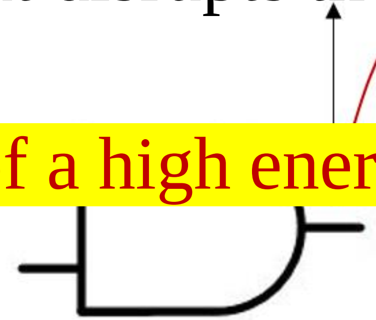
Model checking	Equivalence checking	Theorem proving	SAT/SMT based techniques

What is a Single Event Effect?

- SEEs generally cause no permanent damage (Soft Errors)
    - An electrical disturbance that disrupts the normal operation of a circuit.
    - It is caused by the passage of a high energy particle through a sensitive node in a circuit.

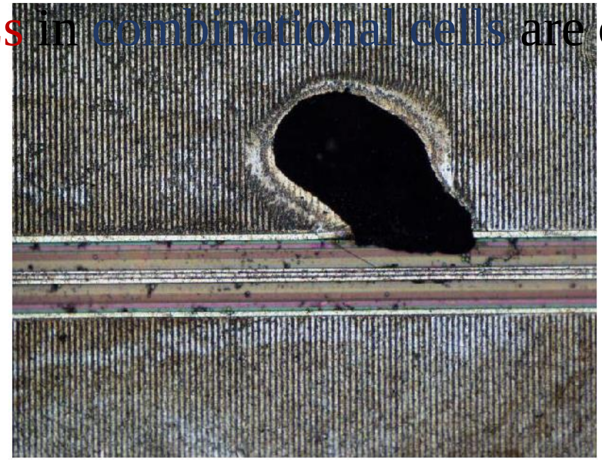
1	1	1
1	0	1
1	1	1

SEU



SET

- SEEs can lead to permanent damage (Hard Errors)
    - SEEs in memory components are called Single Event Upsets.
    - SEEs in combinational cells are called Single Event Transients.

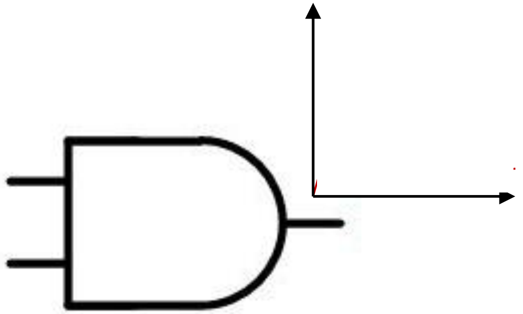


Akturk, A., et al. "Single event effects in Si and SiC power MOSFETs due to terrestrial neutrons." *IEEE Transactions on Nuclear Science* 64.1 (2016): 529-535.

High energy particle

1	1	1
1	0	1
1	1	1

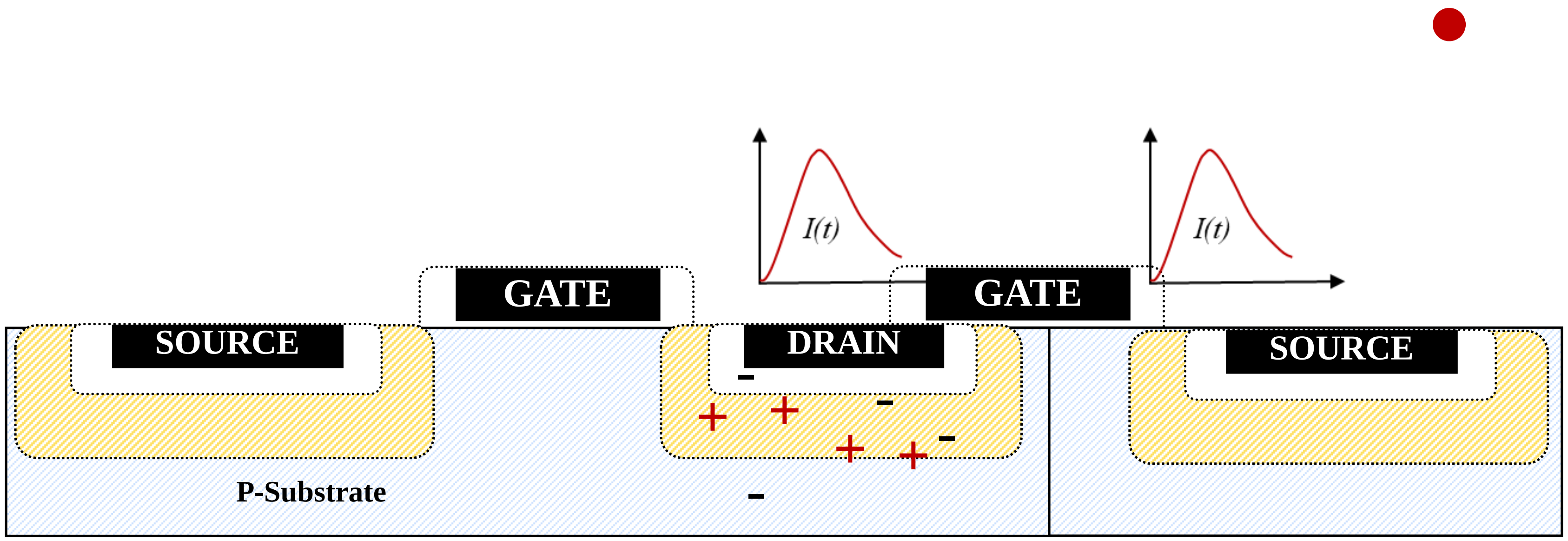
SEU



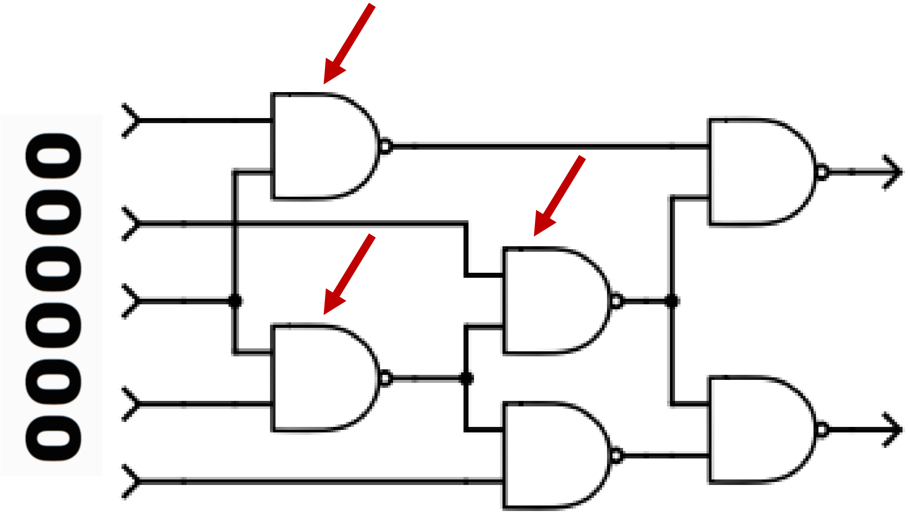
SET

Single Event Multiple Transients

As semiconductor sizes reduces, a single high energy particle strike may induce transients in two logic components



Related Work

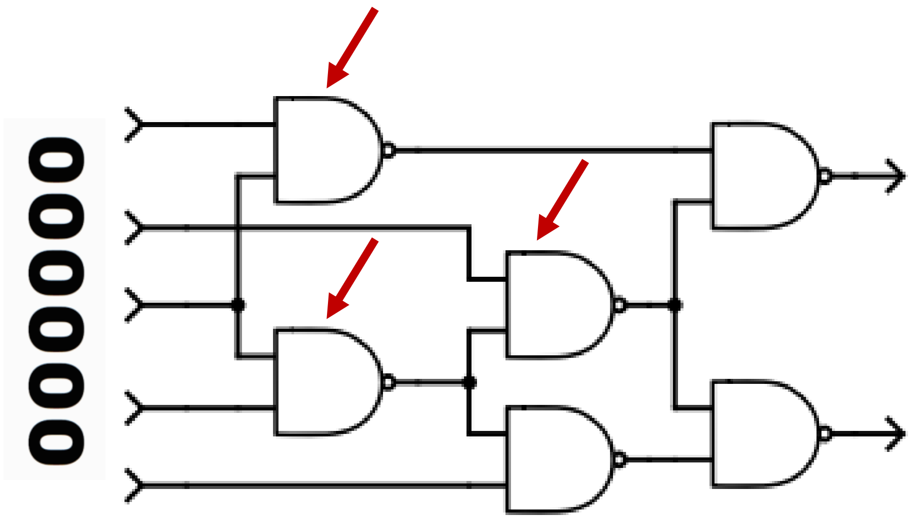


Circuit level fault propagation techniques

- Errors are introduced at Hardware Description Language (HDL) model simulation
- Requires a simple modification to HDL files

Related Work

Circuit level fault propagation techniques



Probability of fault
reaching the output?

5 inputs

$2^5 = 32$ test vectors (for testing ONE fault scenario)

Circuit level fault propagation techniques

10 inputs

$2^{10} = 1024$ test vectors

50 inputs

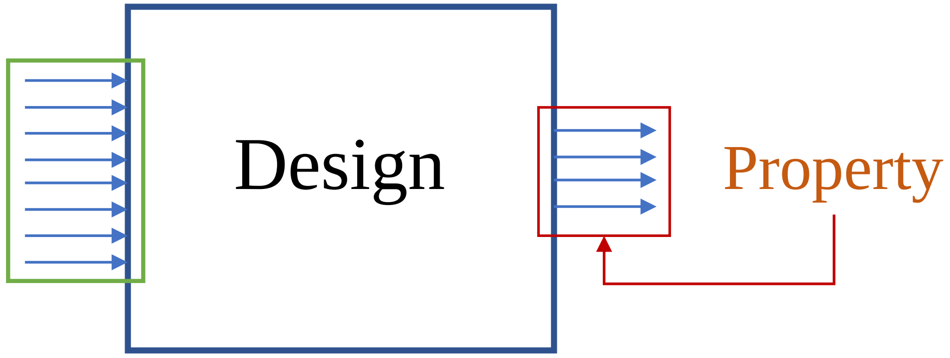
$2^{50} = 1125899906842624$ test vectors (~140 Terabyte)

1125 Trillion (1μs per test vector takes 35 Years)

5 inputs

$2^{5} = 32$ test vectors, for EACH fault scenario

Formal Techniques

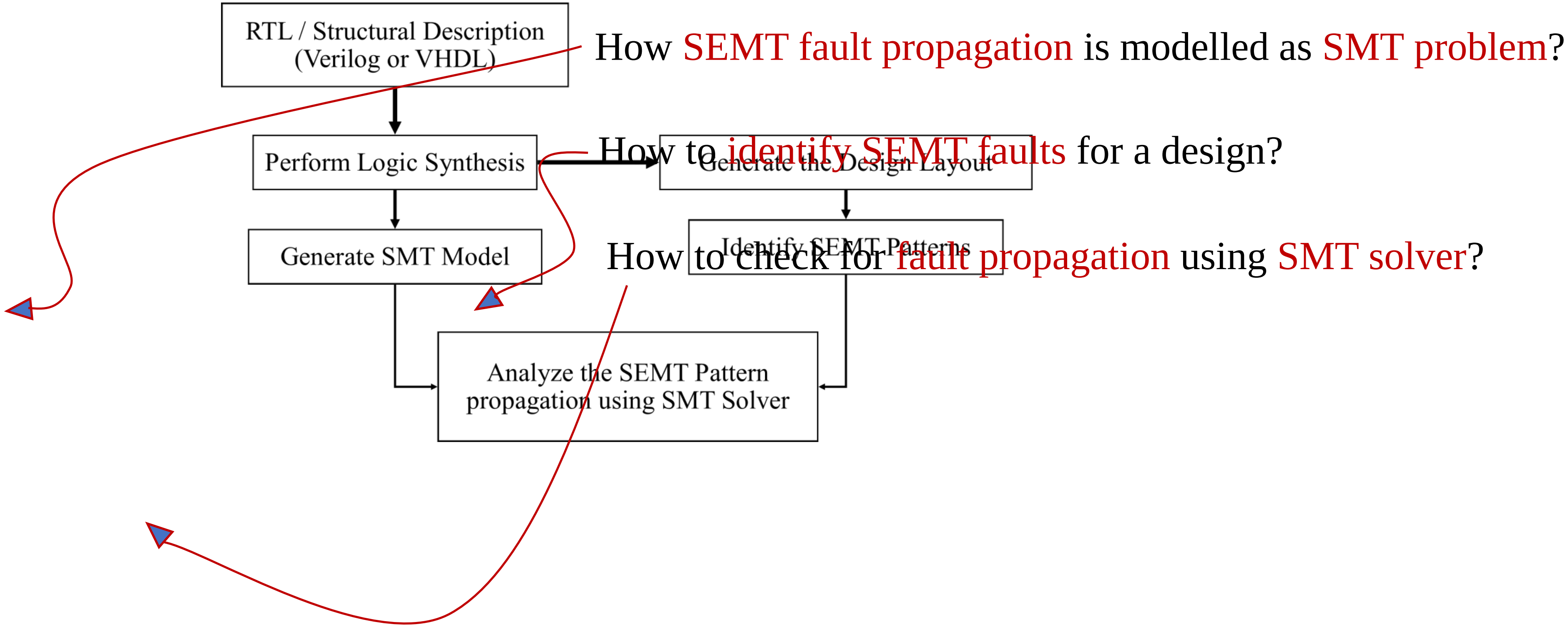


- Check if the property is valid at the output
- Holds for all input vectors

Mathematical / Symbolic model

A symbolic model of design is created using Satisfiability Modulo Theories

Proposed Methodology



Satisfiability Modulo Theories (SMTs)

SMT is an extension to Boolean satisfiability problem (SAT)

P : A and B

Is P satisfiable?

Yes, P is True, when
A = True, B = True

Q: A and $\overline{A}$

Is Q satisfiable?

Q is Un-Sat

SAT is limited to propositional logic (Boolean Formulae)

Satisfiability Modulo Theories (SMTs)

SMTs facilitates the use of arithmetic operators, integer and real variables, in addition to boolean constructs

Plus, Minus, Greater Than, Less Than, If-Then-Else, ...

$P : X + Y = X$ → Is P satisfiable?

Yes, P is Satisfiable, when Y = 0, any value of X

$Q: X + 1 = X$ → Is Q satisfiable?

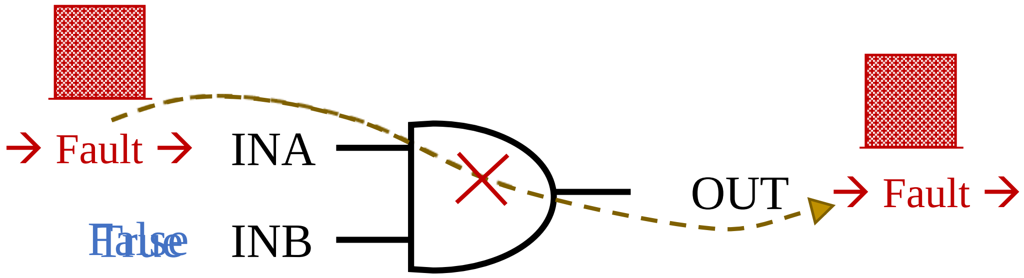
Formula is Un-SAT

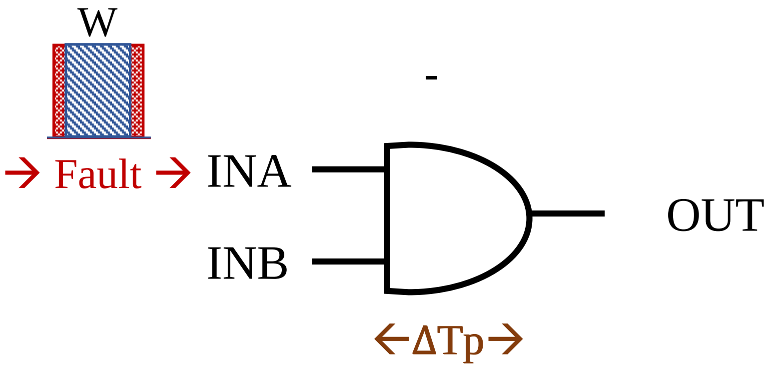
How to analyze the fault propagation using SMT modeling?

SMT modeling of fault propagation in logic gates

Fault propagation is influenced by masking mechanisms

Logical Masking

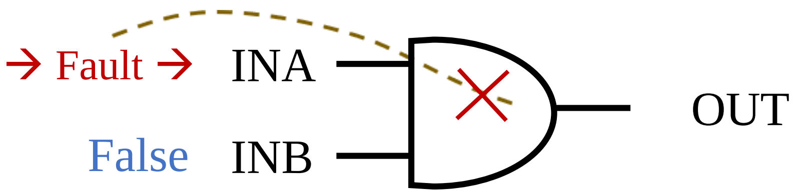




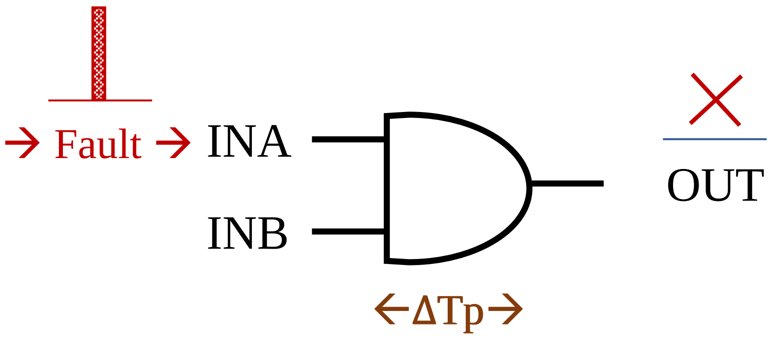
SMT modeling of fault propagation in logic gates

Fault propagation is influenced by masking mechanisms

Logical Masking



Electrical Masking

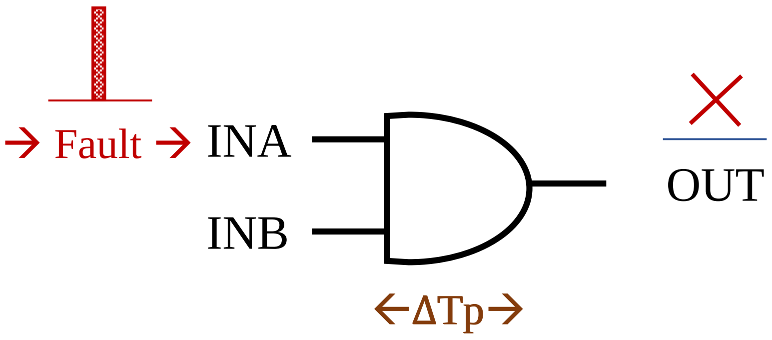


SMT modeling of fault propagation in logic gates

ΔTp = {Rise_INA_Out, Rise_INB_Out, Fall_INA_Out, Fall_INB_Out}

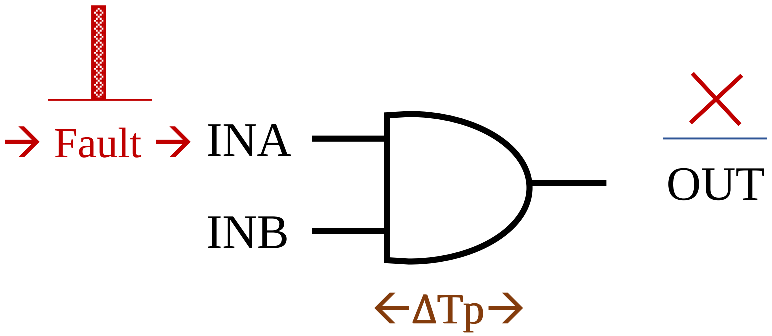
Select the right delay value based on the input signal polarity

Electrical Masking

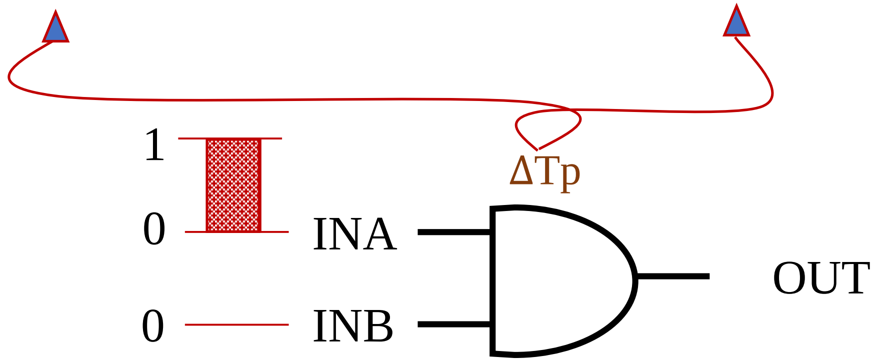


SMT modeling of fault propagation in logic gates

Electrical Masking



ΔTp = {Rise_INA_Out, Rise_INB_Out, Fall_INA_Out, Fall_INB_Out}



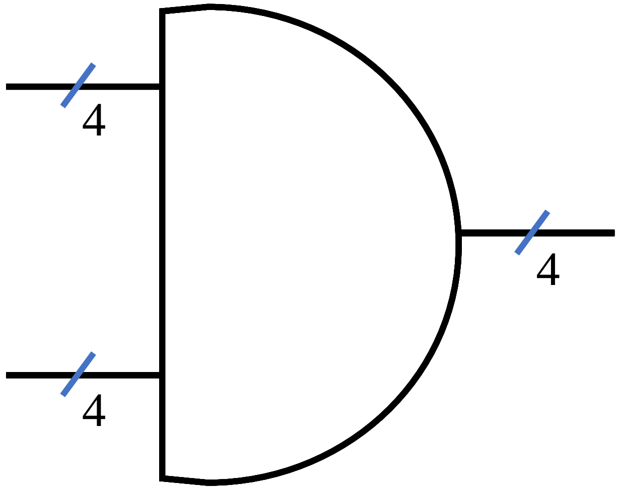
Select the right delay value based on the input signal polarity

SMT modeling of fault propagation in logic gates

Each signal is considered as an array of 4 variables

INA_Faulty
INA_Logic
Prop_Fault_Width_A
Prop_Fault_Time_A

INB_Faulty
INB_Logic
Prop_Fault_Width_B
Prop_Fault_Time_B



Out_Faulty
Out_Logic
Out_Fault_Width
Out_Fault_Time

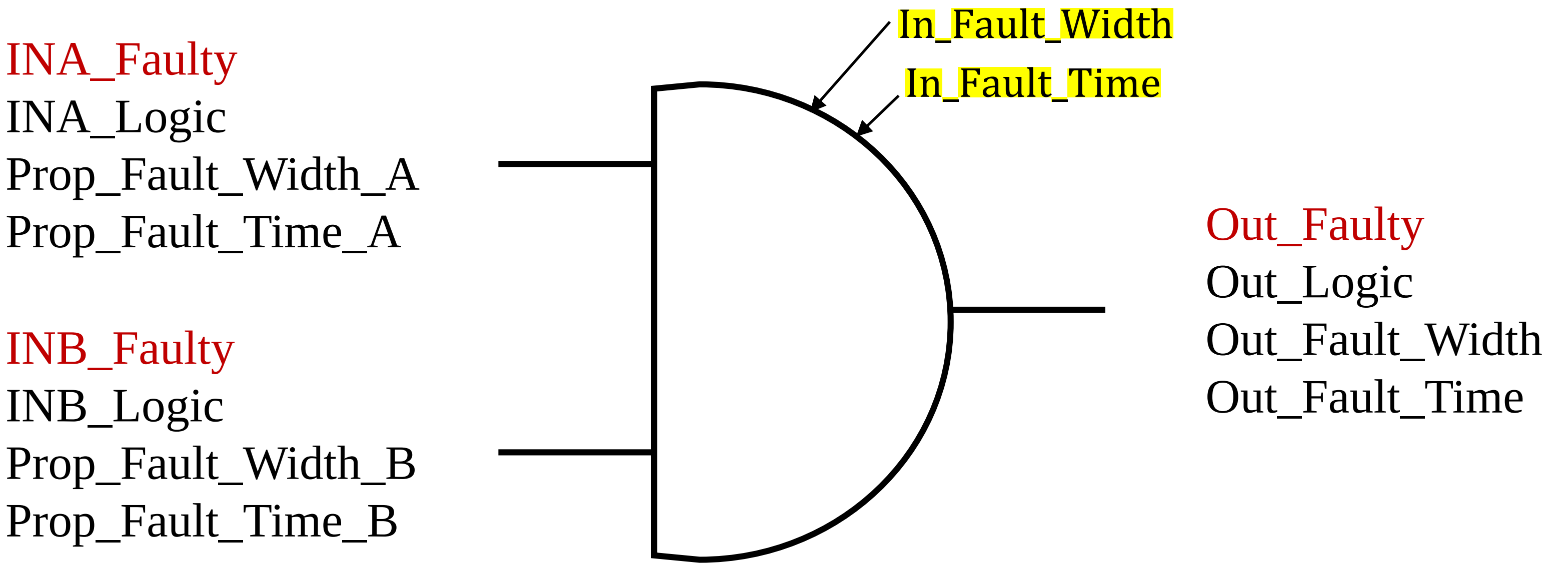
Faulty → Fault presence at the Input → Boolean

Logic → Logic value at the Input → Boolean

Prop_Fault_Width → Width of the fault propagating through the Input → Integer

Prop_Fault_Time → Fault arrival time at the Input → Integer

SMT modeling of fault propagation in logic gates



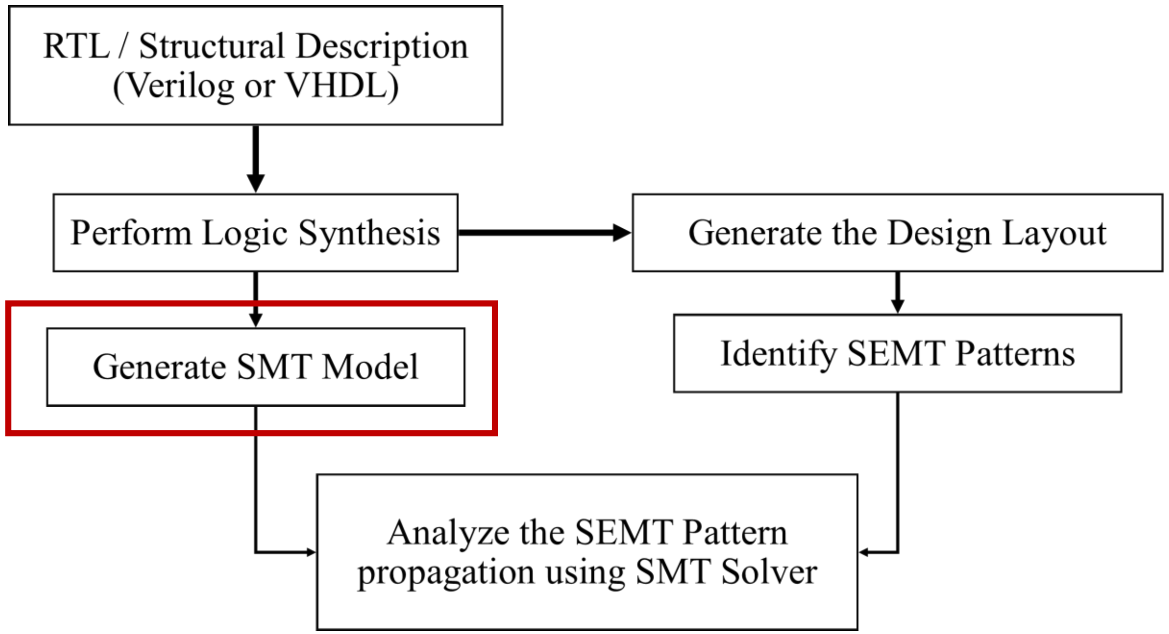
Additional variables are allocated to facilitate the fault injection into a gate

In_Fault_Width → Inject a fault of specific width into the gate

In_Fault_Time → Fault injection time

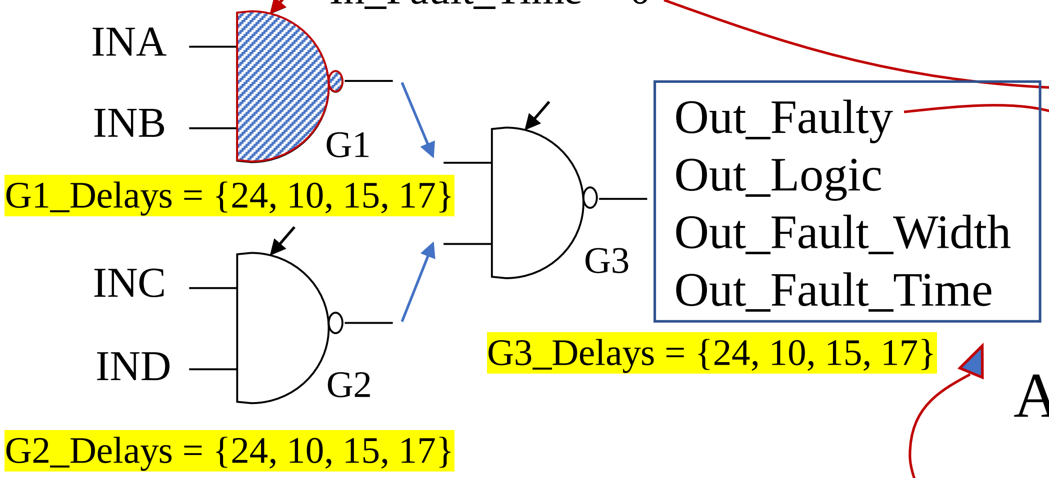
A library of SMT models for all basic gates has been developed

Proposed Methodology



SMT modeling of fault propagation in Combinational circuits

In_Fault_Width = 100
In_Fault_Time = 0



Solver.add_assertion(And(Equals(In_Fault_Width G1, Int(100)),
Equals(In_Fault _ime, Int(0))))

Solver.add_assertion(Equals(Out_Faulty, True))

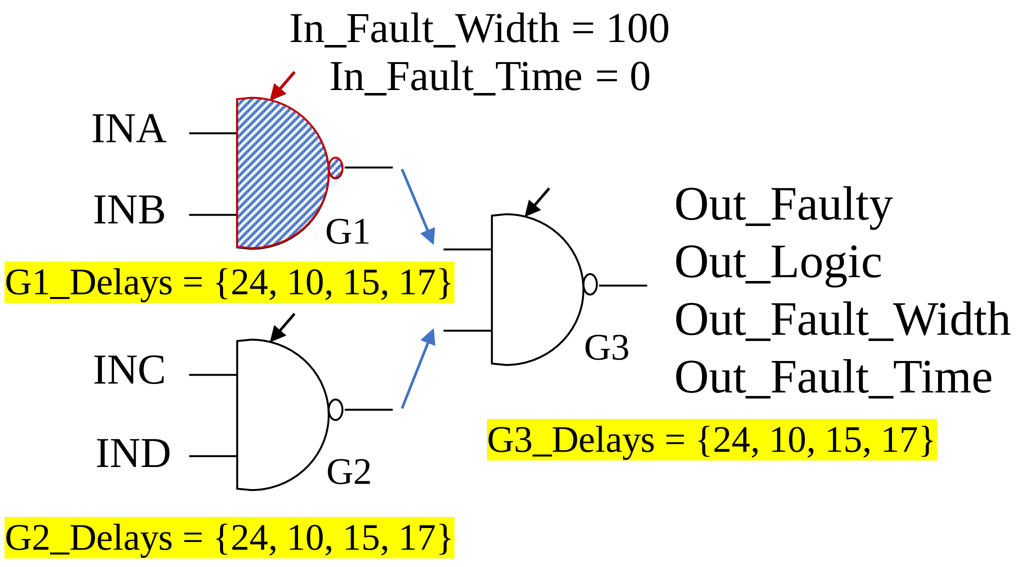
An assertion is satisfiable if it holds at least one model !

Solver.Check_Sat()

An assertion is valid if all models are enumerated!

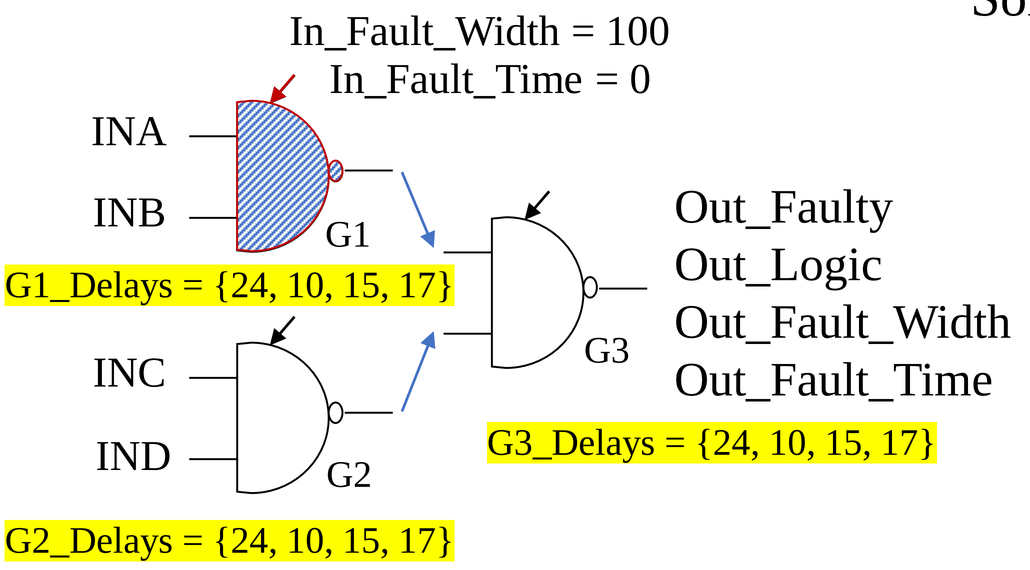
Symbolic Formulae

SMT modeling of fault propagation in Combinational circuits



A	B	C	D	G1	G1_fault	Fault Width	Rise/Fall		
0	0			1	0	100ns	15ns		
0	1			1	0	100ns	15ns		
1	0			1	0	100ns	15ns		
1	1			0	1	100ns	24ns		
0	0	0	0			85ns	15ns		fault propagate
0	0	0	1			85ns	15ns		fault propagate
0	0	1	0			85ns	15ns		fault propagate
0	1	0	0			85ns	15ns		fault propagate
0	1	0	1			85ns	15ns		fault propagate
0	1	1	0			85ns	15ns		fault propagate
1	0	0	0			85ns	15ns		fault propagate
1	0	0	1			85ns	15ns		fault propagate
1	0	1	0			85ns	15ns		fault propagate
1	1	0	0			76ns	24ns		fault propagate
1	1	0	1			76ns	24ns		fault propagate
1	1	1	0			76ns	24ns		fault propagate
0	0	1	1			0ns	0ns		fault masked
0	1	1	1			0ns	0ns		fault masked
1	0	1	1			0ns	0ns		fault masked
1	1	1	1			0ns	0ns		fault masked

SMT modeling of fault propagation in Combinational circuits

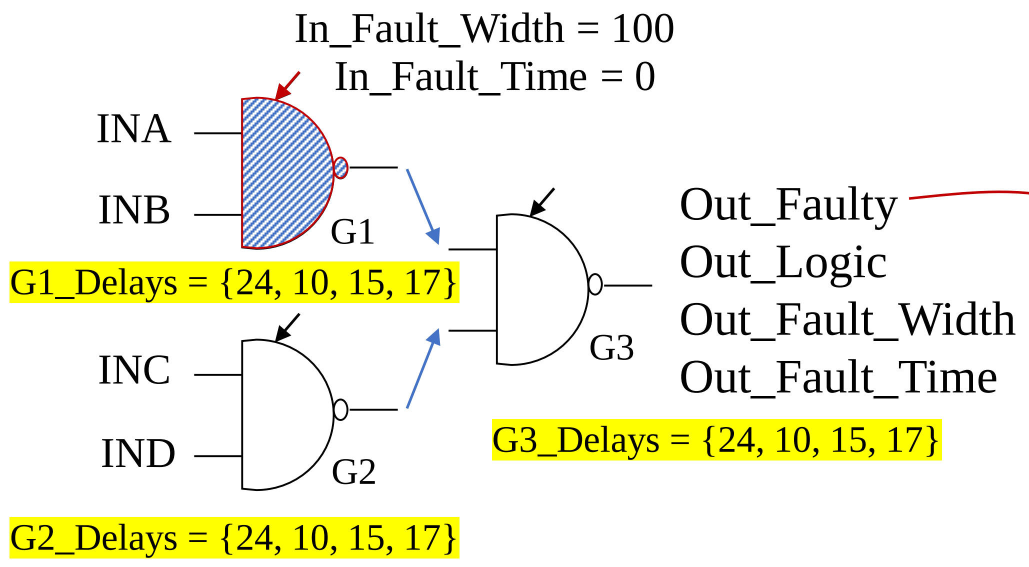


Solver.get_values (INA_Logic, INB_Logic, INC_Logic, IND_Logic, Out_Fault_Width Out_Fault_Time)

```
# Input signal assignments that PROPAGATE the fault
Output Cone : [INA_Logic, INB_Logic, INC_Logic, IND_Logic]
#      INA    INB    INC    IND   Out Fault Width  Out Fault Time
```

Solver.add_assertion(Equals(Out_Faulty, True))

SMT modeling of fault propagation in Combinational circuits



Solver.add_assertion(And(Equals(In_Fault_Width G1,
Int(100)),
Equals(In_Fault_Time; Int(0))))

Solver.add_assertion(Equals(Output_Faulty, False))

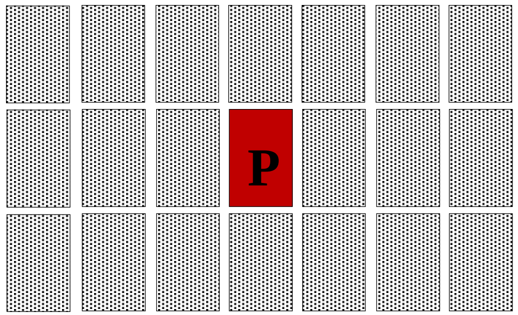
```
# Input signal assignments that MASK the fault
Output Cone : [INA_Logic, INB_Logic, INC_Logic, IND_Logic]
#       INA    INB    INC    IND   Out Fault Width  Out Fault Time
```

SEMT error site identification

Single Event Multiple Transient Analysis → Faults are injected into multiple gates at the same time

Extract the affected area observed in radiation experiments

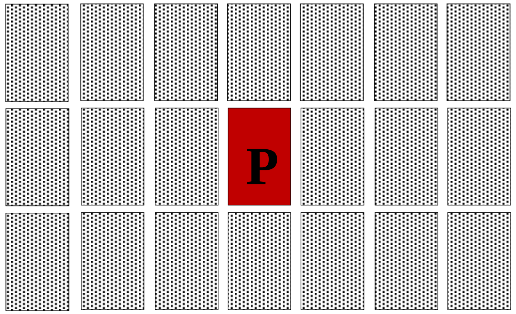
Consider all logic cells covered by a similar surface to be SEMT error patterns



Layout view of test circuit (Chain of Inverters)

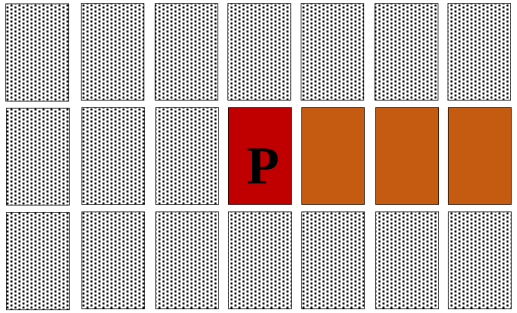
High energy particle strikes a gate P

SEMT error site identification

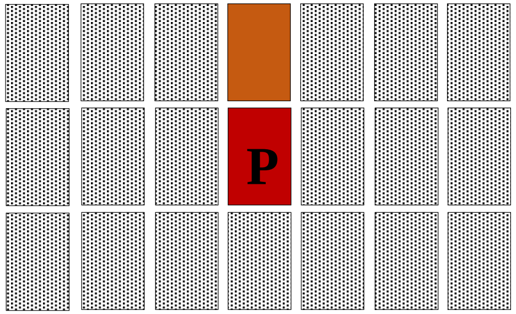


High energy particle strikes a gate P

Layout view of test circuit (Chain of Inverters)



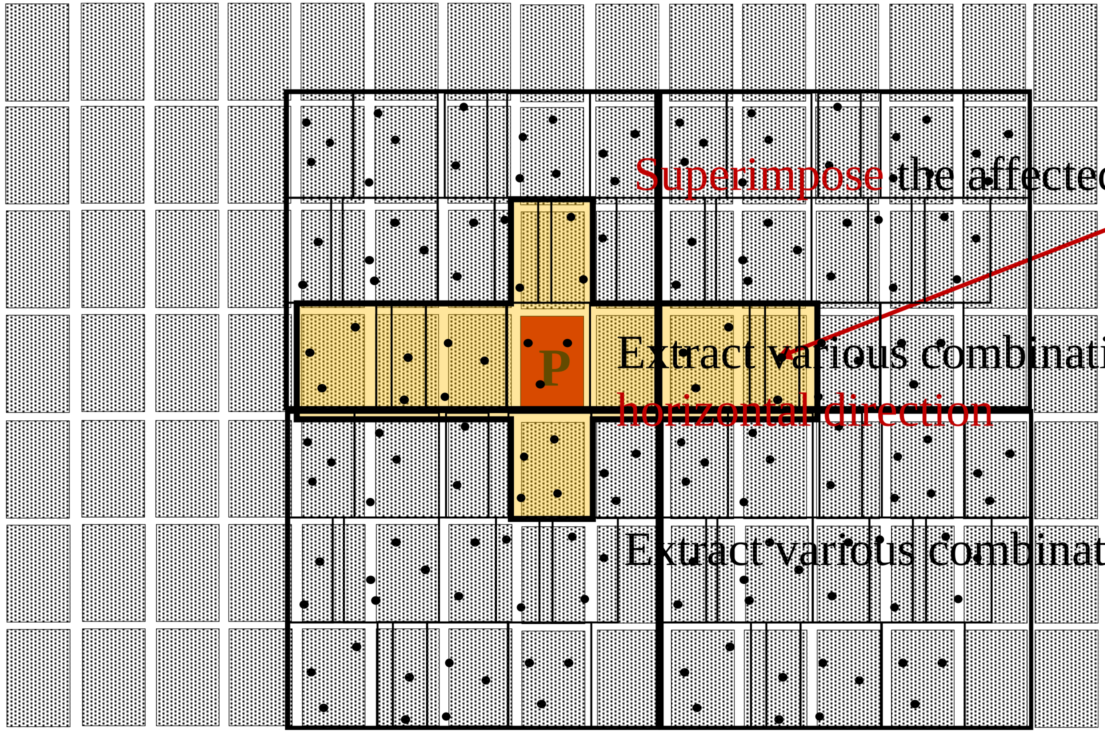
Largest observed error pattern in horizontal direction → SEMT4



Largest observed error pattern in vertical direction → SEMT2

Evans, Adrian, et al. "Single event multiple transient (SEMT) measurements in 65 nm bulk technology." *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. IEEE, 2016.

SEMT error site identification



Superimpose the affected area over the Design Under Analysis

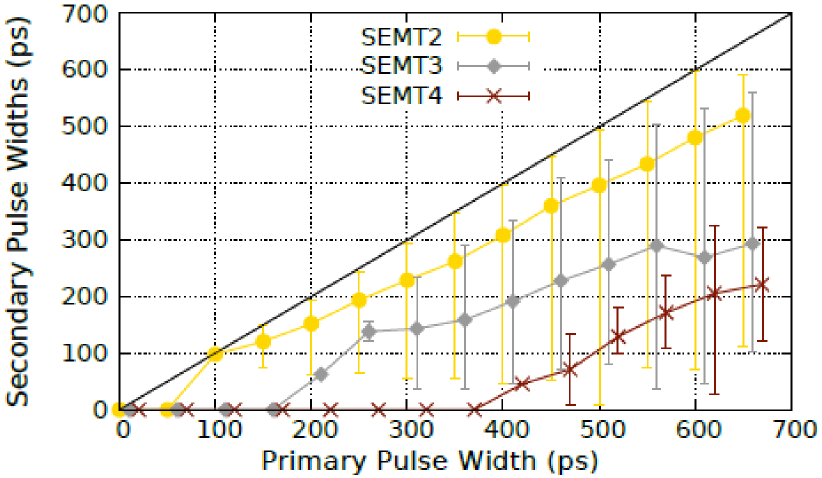
SEMT error range

Extract various combinations of SEMT2, SEMT3, SEMT4 in horizontal direction

Extract various combinatio

Layout view of test circuit (Chain of Inverters)

Layout view of design under analysis

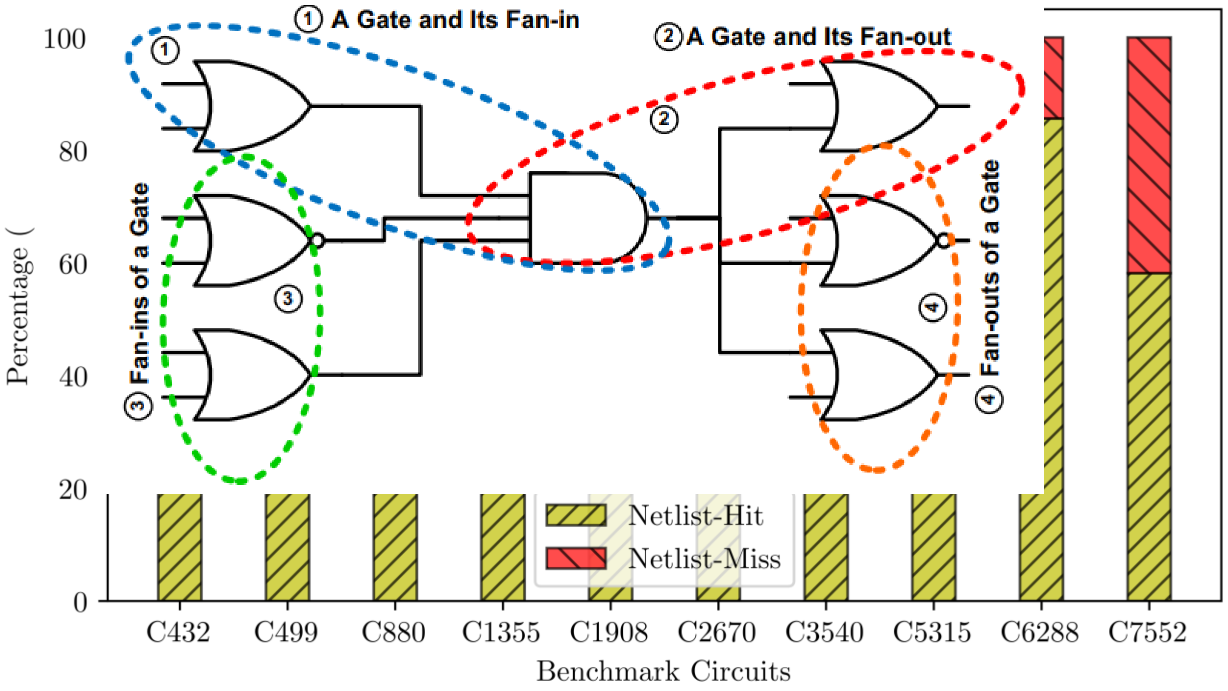


Inject the fault widths according to the pulse width distribution table

Evans, Adrian, et al. "Single event multiple transient (SEMT) measurements in 65 nm bulk technology." *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. IEEE, 2016.

SEMT error site identification

Importance of circuit’s layout for performing fault propagation analysis



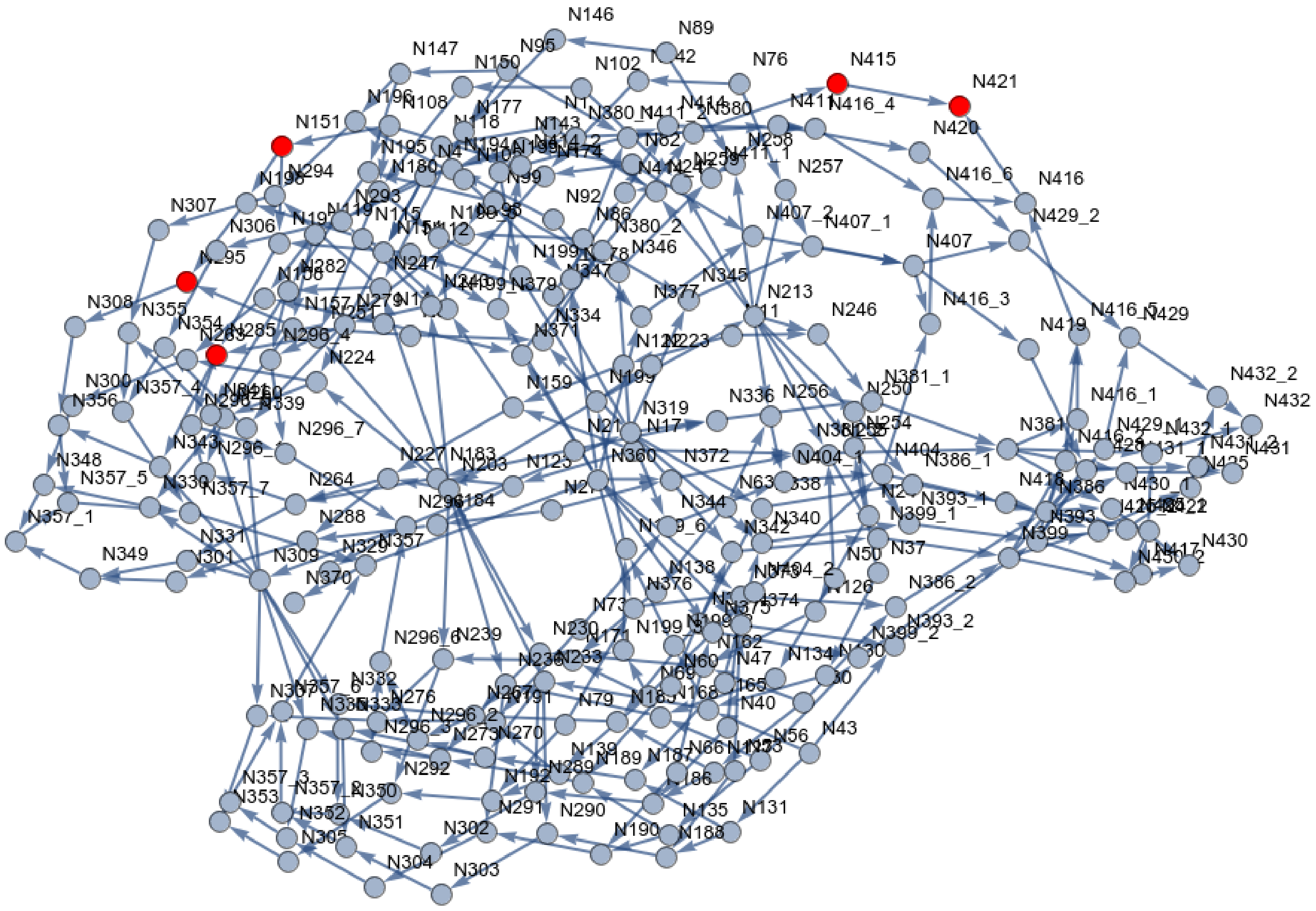
Existing techniques extract SEMT errors from circuit netlist

Not considering layout can miss an average of 50% of the possible SEMT error patterns

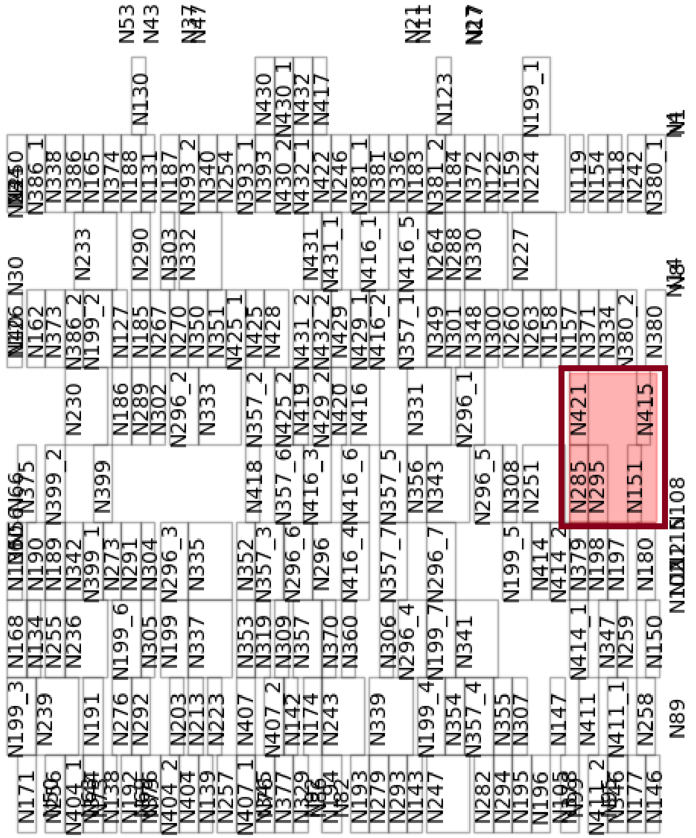
Need layout for accurate error propagation probabilities

Fazeli, Mahdi, et al. "Soft error rate estimation of digital circuits in the presence of multiple event transients (METs)." *2011 Design, Automation & Test in Europe*. IEEE, 2011.

SEMT error site identification

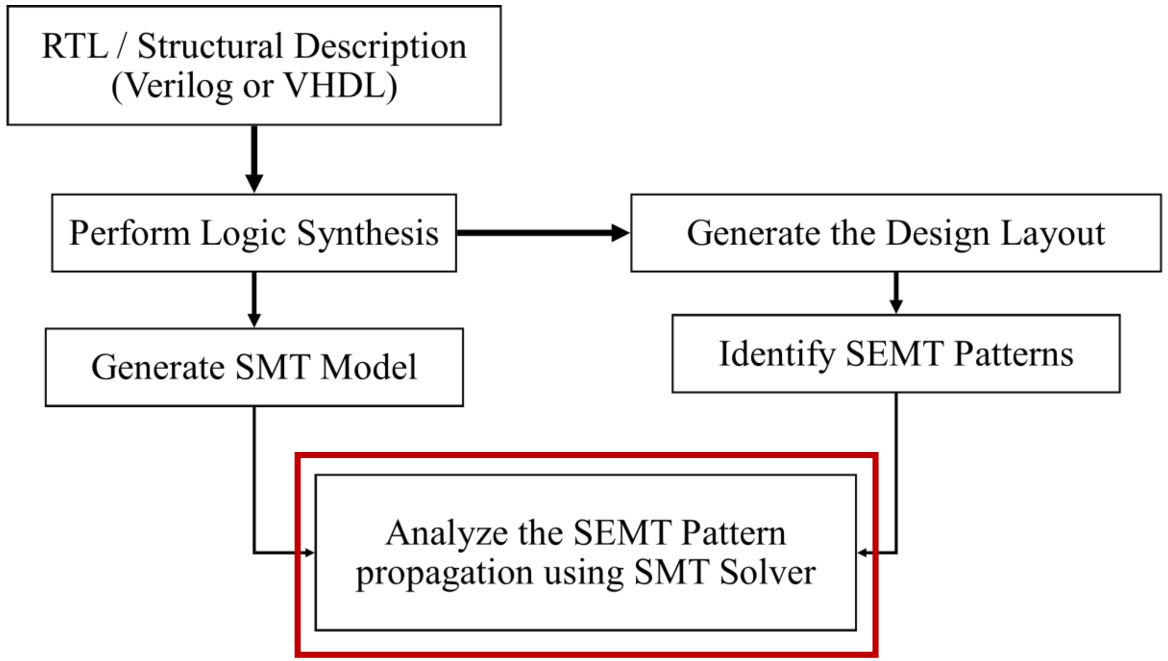


Netlist graph of circuit C432



Layout of circuit C432

Proposed Methodology



Calculating the SEMT error propagation probability

The probability that the injected SEMT fault propagates to the output is computed as

$$P(SEMT_i) = \frac{\{Satisfying\_Input\_Logic\_Assignments\}}{N}$$

$N = 2^n$ For a design with ‘n’ inputs

Soft Error Rate (SER) is used to indicate the vulnerability of digital circuits due to soft errors.

$$SER = \frac{\sum_{i=1}^{i=n} P(SEMT_i)}{n}$$

For ‘n’ SEMT patterns

Calculating the SEMT error propagation probability

$$P(SEMT_i) = \frac{\{Satisfying\_Input\_Logic\_Assignments\}}{N}$$

```
# Input signal assignments that
PROPAGATE the fault
#       INA    INB    INC    IND
```

This process of enumerating the solutions is called Model Counting

SMT Solver assertions stack

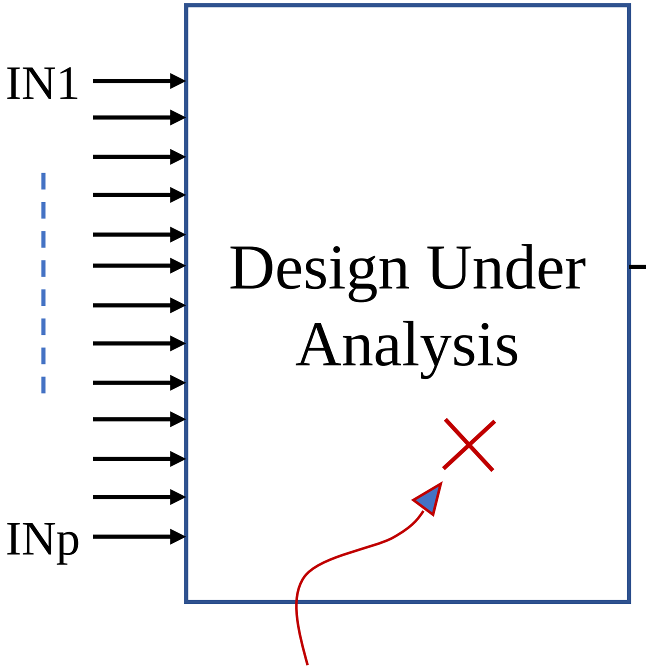
1 by 1 counting is not efficient

Approximate model counters are being developed and can handle smaller formulas

Not feasible when considering the gate delays and electrical masking effects

Calculating the SEMT error propagation probability

We extract probabilities by restricted model counting



Inject SEMT fault

Get_values(IN1, ….., INp)

Randomly select and restrict some of the input signals to values that propagate the injected fault

Restrict the input signals to fault propagating values

IN1	IN2	IN3	…	IN17	…	…	..	INp
Sym	Sym	Sym	Sym	Sym	T	T	F	T

Perform model counting for un-restricted input signals

Process is repeated until all the input signals are covered and the model counts are averaged

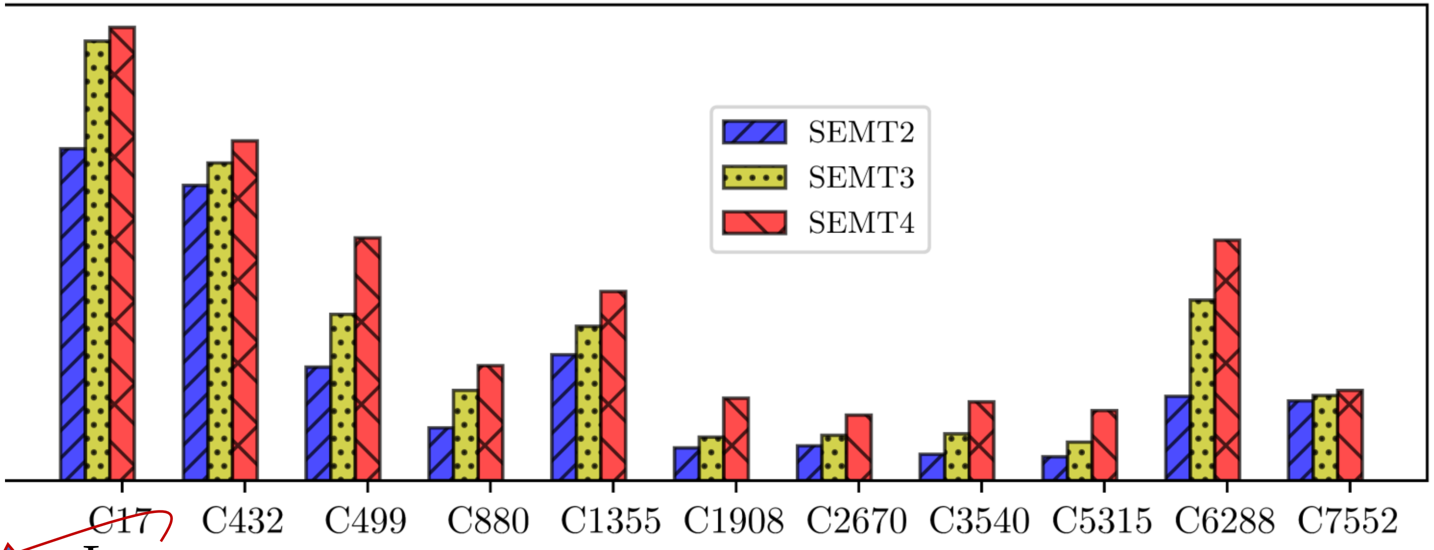
Results

Benchmark	#Gates	#Inputs	#Outputs	Area 45 nm
C17	6	5	2	0.064
C432	160	36	7	2.609
C499	202	41	32	4.473
C880	383	60	26	5.232
C1355	546	41	32	6.682
C1908	880	33	25	11.099
C2670	1269	233	140	18.271
C3540	1669	50	22	23.239
C5315	2307	178	123	38.801
C6288	2416	32	32	27.203
C7552	3513	207	108	49.155

Experiments were performed on ISCAS85 benchmark suite

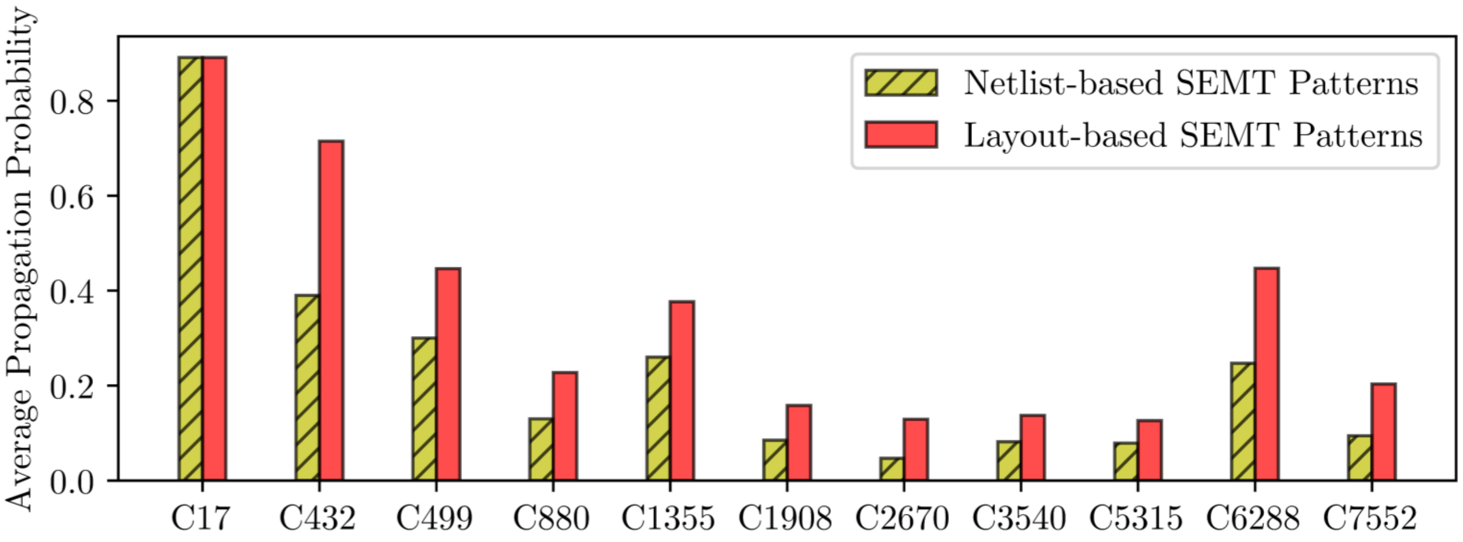
Smallest

All circuits were synthesized to 45nm technology & 65 nm technology



Largest

Results



SEMT fault propagation probabilities when considering the netlist based vs layout based error site identification

Fazeli, Mahdi, et al. "Soft error rate estimation of digital circuits in the presence of multiple event transients (METs)." *2011 Design, Automation & Test in Europe*. IEEE, 2011.

Results

Simulation vs Proposed method

Benchmark	Time (Seconds)			Speed-Up (X)	
	Simulation	Z3	Yices2	Z3	Yices2
C17	5.6	0.23	0.23	24.35	24.35
C432	155.18	6.57	5.19	23.62	29.90
C499	178.32	24.92	19.45	7.16	9.17
C880	298.66	10.5	10	28.44	29.87
C1355	421.35	76.59	63.15	5.50	6.67
C1908	726.11	74.51	59.21	9.75	12.26
C2670	995.32	68.84	60.49	14.46	16.45
C3540	1121.25	139.94	117.2	8.01	9.57
C5315	1433.72	151.04	129.11	9.49	11.10
C6288	1547.56	971.26	743.73	1.59	2.08
C7552	1680.54	248.32	195.42	6.77	8.60

Fault is injected in one version and its output compared to fault free version

Include delay files to consider electrical masking effects

100,000 input vectors

000000

Inject SEMT fault

Fault
Fault
Fault
Fault
Fault

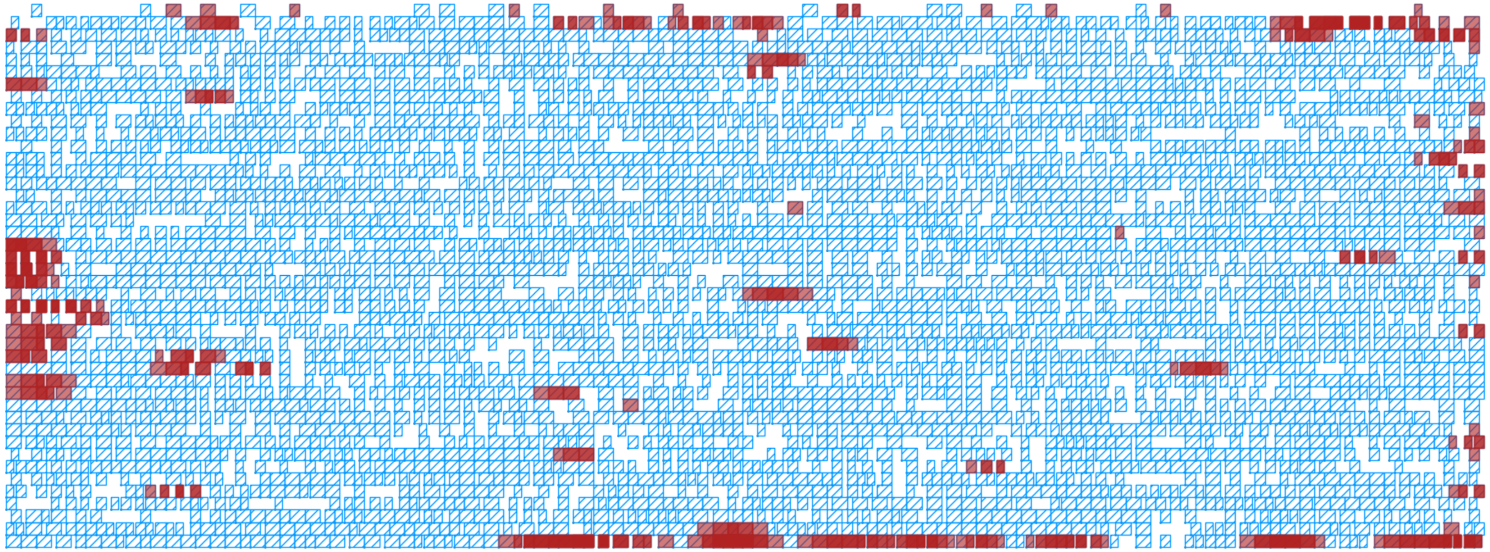
Generate same number of fault propagating models using proposed method.

Results

Fatal SEMT patterns

Design can be checked for fatality against the SEMT faults

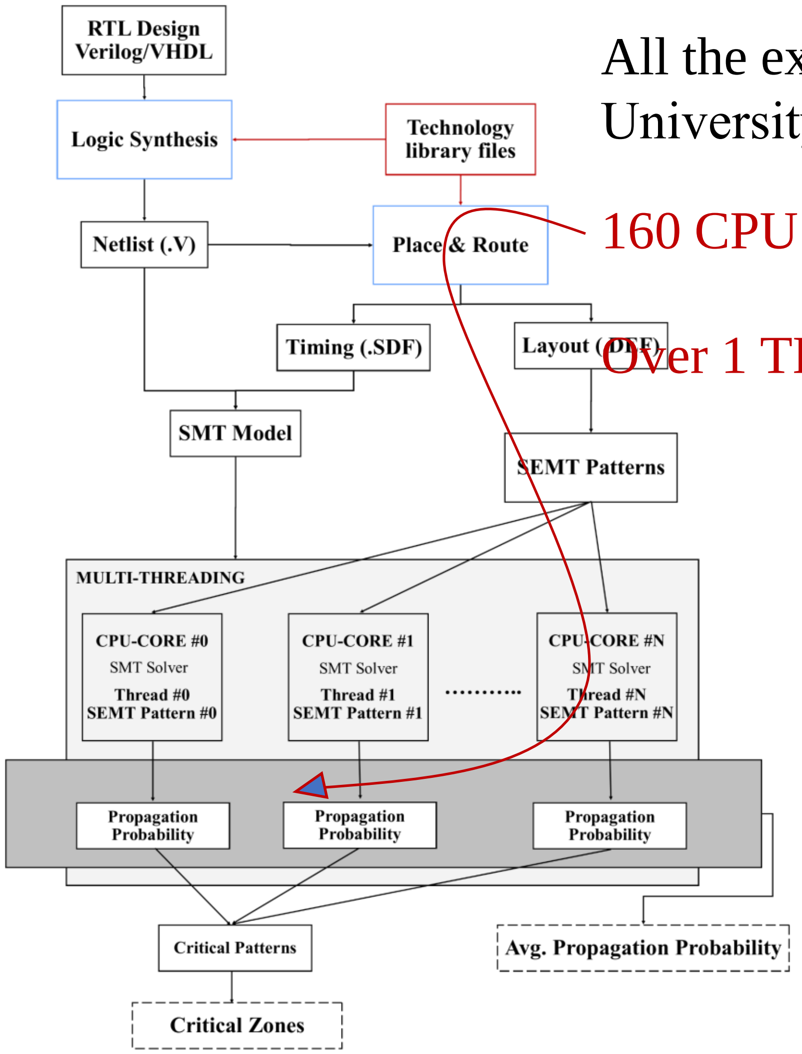
There is no input variable assignment that could mask the fault



Layout of C7552 Gates: 3513, Inputs: 207, Outputs: 108

Such kind of analysis is only possible through formal methods

Multiple Event Transient Analysis Tool - META

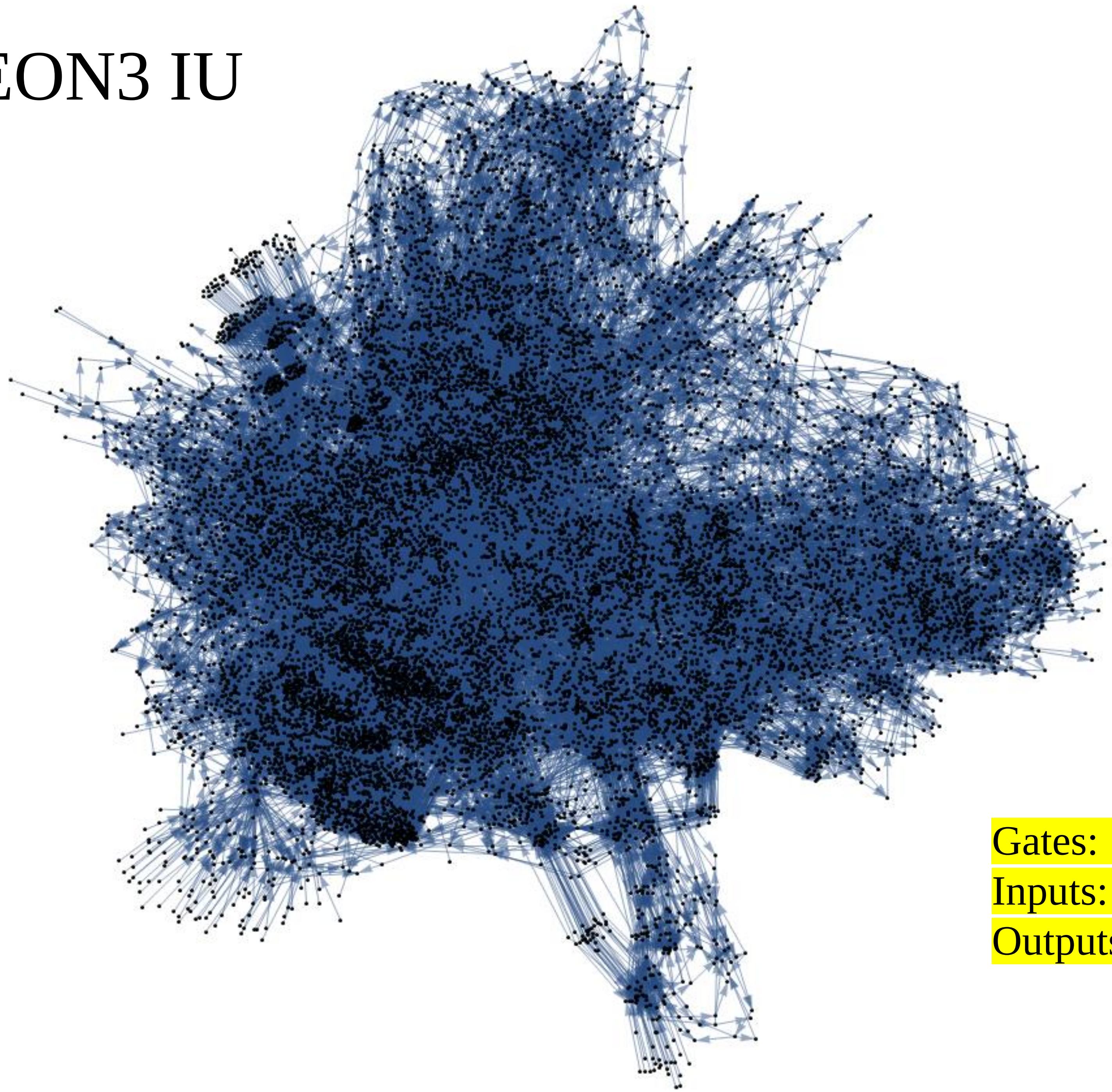


All the experiments were run on Concordia University's CMOS server

160 CPU cores

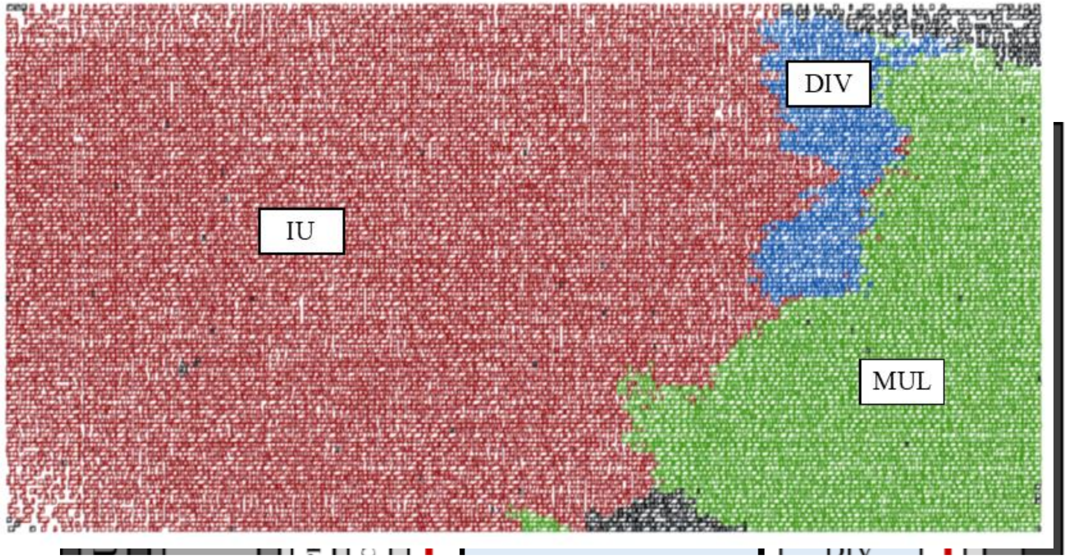
Over 1 TB of main memory

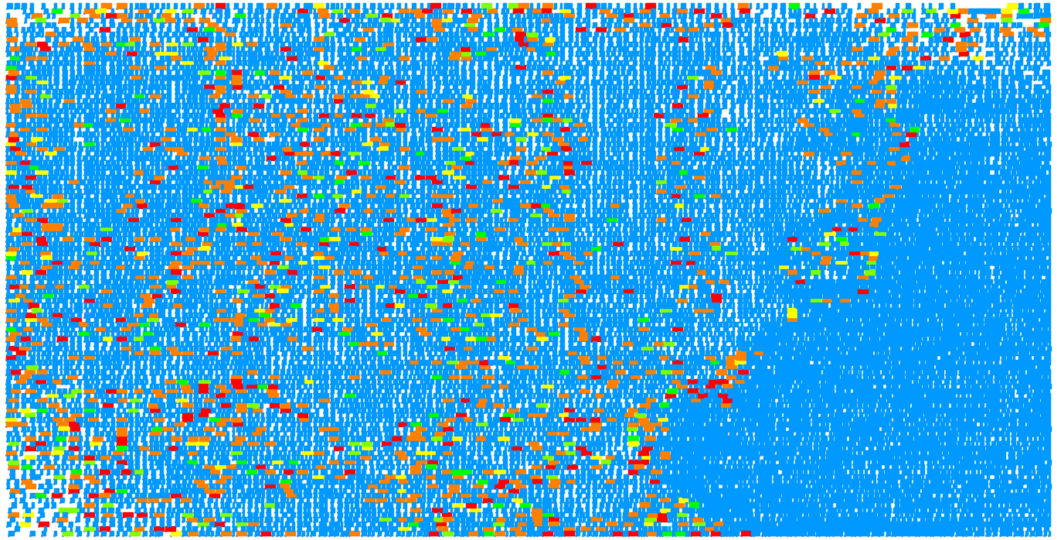
Graph of LEON3 IU



Gates: 12121
Inputs: 1450
Outputs: 2009

SEMT Propagation Analysis on LEON3 Processor Core





Module	Average Propagation Probability			SER
	SEMT2	SEMT3	SEMT4	
IU	0.37	0.43	0.79	0.53
MUL	0.24	0.38	0.50	0.37
DIV	0.26	0.32	0.41	0.33

Modu				Gates
IU				21
MUL	69	66	66	4799
DI				

Module	Time (S)	
	Building SMT Model	SMT Analysis
IU	245.32	5940.23
MUL	89.4	13664.22
DIV	67.3	9442.78

Summary

Modeled SEMT propagation in digital circuits

Layout based SEMT error pattern identification

Identified critical zones over a circuit’s layout

Automated the proposed method into a tool META

For any questions:

otmane.aitmohamed@concordia.ca